



### Sommaire

1	Introduction .....	1
2	Présentation du modèle du cours .....	1
3	La suppression des données .....	2
4	L'insertion des données .....	2
5	La mise à jour de données .....	4

## 1 Introduction

Le cours précédent nous a montré l'ensemble des moyens d'accéder aux données d'une base existante. Néanmoins, SQL ne se limite pas à l'interrogation d'une base. Ce chapitre nous permettra de voir de quelle manière mettre à jour, insérer et supprimer ces données, ce qui est défini par l'acronyme LMD (Langage de Manipulation des Données).

## 2 Présentation du modèle du cours

Pour les exemples nous nous appuyerons sur le même modèle que le cours précédent :

VEHICULE (Code, Type, Marque, Puissance) → les types de véhicule

GARAGE (Code, Nom, Adresse, Ville, Cpostal, Marque) → les garages

PLAQUE (Immat, CodeV#, Nom, Prenom, Adresse, Ville, Cpostal, Date) → véhicules immatriculé

CodeV# de PLAQUE est clé étrangère et référence Code de VEHICULE.

La représentation Type Access

### Le jeu d'essai :

Table VEHICULE

Code	Type	Marque	Puissance
1	Espace	Renault	100
2	Clio	Renault	60
3	Megane	Renault	80
4	Twingo	Renault	50
5	106	Peugeot	45
6	206	Peugeot	60
7	306	Peugeot	80
8	807	Peugeot	110
9	407	Peugeot	130
10	C4	Citroen	85
11	C5	Citroen	100
12	C3	Citroen	45
13	C8	Citroen	115
14	Yaris	Toyota	55
15	Rav4	Toyota	95

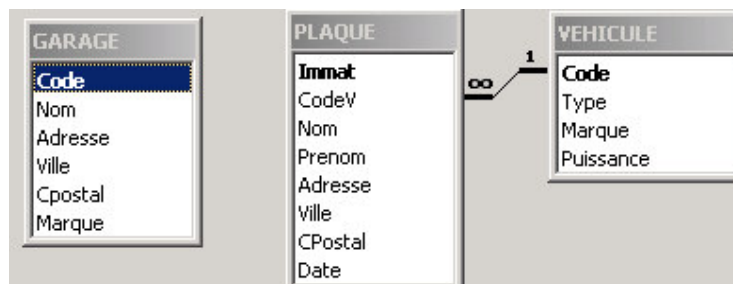


Table GARAGE

Code	Nom	Adresse	Ville	Cpostal	Marque
1	Dupont	36 Route de Paris	Bar Le Duc	55000	Renault
2	Durand	75 Rue de Par là	Nancy	54000	Peugeot
3	Asterix	10 Allée le Ciel	Ailleurs	80100	Renault
4	Obelix	25 Allée du Ciel	Ailleurs	80100	Citroen
5	Le Barde	35 Va en enfer	ICI	81010	Toyota

Table PLAQUE

Immat	CodeV	Nom	Prenom	Adresse	Ville	CPostal	Date
150XY52	1	Lucien	Raymond	36 par ici	Saint Dizier	52000	01/10/2005
200XY55	2	Traban	Fernand	60 rue du lac	Nancy	54000	20/02/2005
300ATX57	3	Tournesol	Luc	80 Avenue de Strasbourg	Metz	57000	18/12/2004
220ACD80	4	Dupond	Louis	35 rue de Moscou	Amiens	80000	20/10/2005
300ATZ81	5	Etienne	Marcel	20 rue de Toulouse	Albi	81990	12/07/2004
400XZ55	6	Durand	Maurice	35 rue de Belfort	Chambley	54120	18/02/2004
20AAA80	7	Lenoir	Louis	70 Rue de Bellefontaine	Amiens	80000	27/05/2005
2875ATX57	8	Leblanc	Lucien	80 Route de Vienne	Albi	81990	12/07/2004



### 3 La suppression des données

Cette suppression est assurée par la commande SQL DELETE. Cette commande permet d'effacer un ou plusieurs enregistrements.

La syntaxe est :

```
DELETE FROM <nom table> WHERE <condition> ;
```

La Clause WHERE permet d'effectuer une restriction sur les enregistrements à effacer.

**Ex :**

```
DELETE FROM GARAGE ;
```

Efface tous les enregistrements de la table GARAGE

```
DELETE FROM GARAGE  
WHERE Marque = "Renault" ;
```

Efface tous les enregistrements de GARAGE dont la marque est Renault

Les clauses WHERE peuvent reprendre toutes les formes vues précédemment

Ex :

Effacer tous les enregistrements de la table véhicule qui sont immatriculés (Cela permet de ne garder dans la table VEHICULE que les véhicules disponibles):

```
DELETE FROM VEHICULE  
WHERE Code IN  
(SELECT CodeV FROM PLAQUE) ;
```

**Attention, cette requête est impossible du fait de l'intégrité référentielle:  
CodeV de PLAQUE est référencé par Code de VEHICULE.**

Il est par contre possible d'effacer les enregistrements non référencés dans la table PLAQUE :

```
DELETE FROM VEHICULE  
WHERE Code NOT IN  
(SELECT CodeV FROM PLAQUE) ;
```

### 4 L'insertion des données

#### 4.1 *Insertion simple :*

Cette insertion est assurée par la commande SQL INSERT. Elle permet d'insérer de nouveaux enregistrements (un ou plusieurs) dans une table.

La syntaxe est :

```
INSERT INTO <nom_table> (liste champs) VALUES (liste valeurs )
```

- Liste champ : nom des champs pour lesquels on fournit une valeur. Inutile si on valorise tous les champs de la table.
- Liste valeurs : valeurs des champs figurant dans la liste des champs (même ordre). Les chaînes de caractères sont délimitées par des guillemets.

**Ex :**

Insertion dans la table Garage d'un enregistrement complet :

```
INSERT INTO GARAGE
VALUES ( 6, "Pere-noel", "chemin des Rennes", "Au pole",
"00000","Traineau");
```

Insertion dans la table Garage d'un enregistrement partiel en spécifiant les champs :

```
INSERT INTO GARAGE
(Code, Nom, Marque)
VALUES ( 7, "Un autre","BMW");
```

Les champs peuvent également être spécifiés dans le désordre :

```
INSERT INTO GARAGE
( Marque, Nom, code)
VALUES ( "BMW", "Un autre", 7);
```

Les champs non connus peuvent également être remplacés par la valeur Null :

```
INSERT INTO GARAGE
VALUES (8, "Un autre",Null ,Null ,Null ,"BMW");
```

#### **4.2 Insertion multiple**

L'insertion multiple consiste à insérer en une seule opération plusieurs enregistrements dans une table. Cette opération peut être réalisée de deux façons :

- Insertion multiple de valeurs :

```
INSERT INTO VEHICULE
VALUES (
(16, "Golf", "Volkswagen", 90),
(17, "Polo", "Volkswagen", 45),
(18, "Passat", "Volkswagen", 115)
);
```

Cette syntaxe ne fonctionne que sous de rare SGBDR (Oracle, ??), la seule solution étant alors d'exécuter plusieurs ordres Insert.



- Insertion multiple à partir d'une commande Select. La syntaxe est alors :

```
INSERT INTO <nom_table> (liste champs) SELECT (requête SQL)
```

La requête SQL doit renvoyer les champs compris dans la liste des champs spécifiés. Si la liste des champs n'est pas spécifié, tous les champs de la table sont considérés.

**Ex :**

```
INSERT INTO PLAQUE (Immat, CodeV)
SELECT "Immat" & Code, Code FROM VEHICULE
WHERE Code NOT IN
(SELECT CodeV FROM PLAQUE);
```

	BTS IG 1 <sup>ère</sup> année ALSI	Chapitre 9 - Cours	 LYCÉE COLLEGE RAYMOND POINCARÉ SARL LE DUC
	<i><b>La manipulation des données</b></i>		



Attention, la commande INSERT ne peut être utilisée pour mettre à jour un enregistrement. En cas d'utilisation sur un champs déjà référencé (clé existante), une erreur sera générée par le SGBDR.

## 5 La mise à jour de données

Cette mise à jour est assurée par la commande SQL UPDATE. Elle permet de mettre à jour un ou plusieurs champs dans un ou plusieurs enregistrement d'une table.

La syntaxe est :

```
UPDATE <nomtable>
  SET <nomchamp1> = expr1, <nomchamp2> = expr2 ...
  [WHERE condition]
```

- SET nomchamp = expr : permet de donner une nouvelle valeur à un champ, expr étant une constante, un autre attribut ou une expression numérique. Expr peut aussi être une commande SELECT entre parenthèse.
- WHERE condition : condition qui détermine les lignes auxquelles s'appliquent la mise à jour.

**Ex :**

```
UPDATE VEHICULE
  SET Puissance = Puissance*10;
```

```
UPDATE VEHICULE
  SET Puissance = Puissance*10
  WHERE Marque ="Renault";
```

Mise à jour de plusieurs attributs :

```
UPDATE VEHICULE
  SET Puissance = Puissance*10, Marque = Marque & "*10"
  WHERE Marque ="Renault";
```

```
UPDATE PLAQUE
  SET Nom = UPPER(Nom); (UCASE pour Access)
```

Pour la requête suivante, modification du MRD :

VEHICULE (Code, Type, Marque, Puissance, NomV)

```
UPDATE VEHICULE
  SET NomV = (SELECT Nom FROM PLAQUE WHERE Code = CodeV)
  WHERE Code IN (SELECT CodeV FROM PLAQUE);
```

Attention, le SELECT dans un UPDATE ne fonctionne pas sous Access.

Cette requête peut également être réalisée par une jointure :

```
UPDATE VEHICULE, PLAQUE
  SET NomV = Nom
  WHERE Code = CodeV AND
  Code IN (SELECT CodeV FROM PLAQUE);
```